# SPECIFICATION

Electronic Version 1.2.8

Stylesheet Version 1.0

# DESIGN AND IMPLEMENTATION DEVICE FOR REAL-TIME CONTROLLERS

## Background of Invention

[0001]     The present invention relates generally to medical imaging systems, and more particularly to an apparatus and method for implementing pre-designed state models for operation of real-time controllers within an X-ray imaging system.

[0002]     X-ray imaging systems contain real-time controllers, which are built to control distributed hardware components within an X-ray imaging system. The hardware components are controlled in real-time. X-ray imaging systems are used for various applications having varying degrees of complexity. For example, an X-ray application such as a Digital Subtraction Angiography (DSA) is considered a more complex application and is performed by coordinating the operation of multiple real-time controllers.

[0003]     Real-time controllers vary in scope depending upon the application, the X-ray imaging system, and the function to be performed. They are designed and implemented independently to perform the various functions such as X-ray generation, patient positioning, image acquisition, and image processing. Real-time controllers are required to provide coordination of runtime activities, and may be required to satisfy timing and performance constraints such as in Vascular and Radiography and Fluoroscopy X-ray applications.

[0004]

Behavior of real-time controllers is traditionally specified as a state machine, which can be described by a state model. State machines describe how a real-time controller responds to stimuli received from the X-ray system in the form of events.

The state machines are implemented using handwritten code. Writing of the code is a labor-intensive process that tends to be error-prone. Also, it is very difficult to ensure that the implementation of the state machines matches designed specifications.

[0005]     Therefore, it would be desirable to provide an apparatus and method for implementing of state machines that is less labor-intensive than the current art and is also less error-prone. The apparatus and method may provide a more accurate implementation as to better match design specifications.

## Summary of Invention

[0006]     The foregoing and other advantages are provided by an apparatus and method for implementing pre-designed state models for operation of real-time controllers within an X-ray imaging system. A method for implementing a pre-designed state model is provided including extracting state information from the state model. The extracted state information is processed to generate a state code and a state table. The state code is compiled to generate a runtime code. The runtime code is used in combination with the state table to implement the state model.

[0007]     A state processor for generating a state table and a runtime code for use in implementing of one or more pre-designed state models is also provided. The state processor includes a state model information provider, a state information separator, and a compiler. The state model information provider extracts state model information in response to the one or more state models. The state information separator generates a state code and the state table in response to the one or more state models. The compiler compiles the state code and generates the runtime code.

[0008]     One of several advantages of the present invention is that it provides designers with versatility as to alter and automatically update code at any time.

[0009]     Another advantage of the present invention is that use of a state model provides hierarchically organized states, in turn providing improved scalability and understandability of state models.

[0010]
        Furthermore, generic components are provided that implement event handling and state transitions. This allows state machines to be modified without altering basic

functions performed by the controller.

[0011]    Moreover, the present invention provides communication and cooperation between separate state machines. This also provides increased versatility.

[0012]    The present invention through the use of state model diagrams and an automated process provides increased performance and is less error prone.

[0013]    The present invention itself, together with attendant advantages, will be best understood by reference to the following detailed description, taken in conjunction with the accompanying figures.

## Brief Description of Drawings

[0014]    For a more complete understanding of this invention reference should now be had to the embodiments illustrated in greater detail in the accompanying figures and described below by way of examples of the invention wherein:Figure 1 is a block diagrammatic view of a X-ray system using a state processor in accordance with an embodiment of the present invention; Figure 2 is a process flow diagram illustrating a state processor process of the present invention, for implementing of a state model, in accordance with an embodiment of the present invention; and Figure 3 is an example of an implementable state model diagram according to an embodiment of the present invention.

## Detailed Description

[0015]    While the present invention is described with respect to a method and apparatus for implementing pre-designed state models for operation of real-time controllers within an X-ray imaging system, the present invention may be adapted for use in various systems including: radiotherapy systems, X-ray imaging systems, ultrasound systems, nuclear imaging systems, computer based systems, and other systems that use state models.

[0016]    In the following description, various operating parameters and components are described for one constructed embodiment. These specific parameters and components are included as examples and are not meant to be limiting.

[0017]    Referring now to Figure 1, a block diagrammatic view of an X-ray system 8 using a state processor 10 in accordance with an embodiment of the present invention is shown. Real-time controllers 11 are a cooperating set of controllers that control distributed hardware components 12 in real-time. Examples of hardware components may be an X-ray generator 13, an image acquisitioning device 14, an image processor and display 15, and a patient positioning device 16.

[0018]    The hardware components 12 are controlled by a series of events. The run-time controllers 11 implement the events in accordance to a pre-designed cooperating set of state models 17. The term "cooperating" is referring to the manner as to which the state models are interlaced and operate together to perform various functions, this will become more evident in the following description. The state models 17 are manually entered into a state model information provider 18 by an operator. The state models 17 include multiple state models each of which having multiple events. The device 10 generates state tables 18 and runtime codes 19 that are used by the run-time controllers 11 as to implement the multiple events through the use of three software-based tools 20, a runtime library 21, and a compiler 22. Although each hardware component 12 is shown as having a single run-time controller 11, each hardware component 12 may have more than one run-time controller. Also, each run-time controller 11 has its own state table 18 and runtime code 19 associated with actions for that corresponding hardware component 12. The three software-based tools 20 are the state model information provider 24, a state information separator 26, and an event organizer 28.

[0019]    The state model information provider 24 is a software-based program that extracts state model information from the state models 17, such as event information and transitional information between each event. The state model information is converted to text format and transferred to the state information separator 26.

[0020]
          The event organizer 28 is a software-based program that generates an events symbols header from an events configuration file and a header file, and is best shown in Figure 2. Although the event organizer 28 of the present invention is written in the Perl programming language other similar programming languages known in the art may be used. The events configuration file and the header file define events to be

shared by the state models 17. The events symbols header contains global and shared event symbol definitions. Recognizing shared or common events between the state models 17 prevents duplication of stored events. The events symbol header contains a centralized list of all events for easy addition or renaming of events.

[0021]    The state information separator 26 is also a software-based program that processes the events symbols header in combination with the state model information to generate state codes and the state tables 18, also best shown in Figure 2.

[0022]    The runtime library 21 contains a library of components including a dynamic event processor 30, a time and memory efficient interpreter 32, and a scripted dynamic event processor 34. The dynamic event processor 30 contains the definitions and information needed to implement events. The dynamic events processor 30 also provides the framework for handling of the events. The time and memory efficient interpreter 32 processes events and performs actions according to information contained in the state table. Although, the runtime library 21 is designed to meet timing and performance requirements of X-ray imaging applications, it may be designed to meet timing and performance requirements of other systems. The scripted dynamic event processor 34 may be used to annotate the state models using a script programming language known in the art, such as Tcl. The annotation of state models allows state behavior to be changed without having to rebuild the runtime code.

[0023]    The compiler 22 compiles the state code using the events symbols header and the components contained within the runtime library 21 to produce a runtime code. The runtime code may be perceived as a state machine in that it is the code that is used to control the hardware components 12. The compiler 22 may be a C++ compiler or other compiler known in the art.

[0024]    The real-time controller 11 may be microprocessor based controllers such as computers having central processing units, memory (RAM and/or ROM), and associated input and output buses. The controllers 11 implement the events in the set of state models as instructed by the runtime codes while referring to the corresponding state tables. The state tables are used to provide the state behavior expected from the controllers 11. Although, a single controller 11 is shown for each

hardware component 12, a single main controller may be used in replacement of the controllers 11 as to control the hardware components 12.

[0025]    Referring now to Figure 2, a process flow diagram illustrating a state machine process of the present invention, for implementing of a state model by a single run-time controller 11, in accordance with an embodiment of the present invention is shown. In the following diagram the solid lined arrows represent the flow of transitional steps, the dashed lined arrows represent flow of information including generated files, code, and data.

[0026]    In step 50, an operator inputs one or more state models 52, which may be visible, through the use of a drawing tool (not shown) into the state model information provider 24. State model drawing tools are known in the art. The state model information provider 24 extracts state information 54, converts the information 54 to text format, and transfers the information 54 in text format to the state information separator 26. In doing so, the state model information provider 24 determines what states and events exist in the state model and the different transitions between each state.

[0027]    In step 56, as stated above the events symbols header 60 is generated from an events configuration file 62 and the header file 64, via the event organizer 28. The events symbols header 60 contains global and shared event symbol definitions used, in a forthcoming step, for compiling of the state code.

[0028]    In step 66, the state information separator 26 using the state model information 54 and the events symbols header 60 generates the state code 68 and the state table 18. The state table 18 contains information to be used by a real-time controller 11 at run-time to perform state transitions and to identify the actions and conditions associated with the transitions. The state code 68 contains software code, such as C++, that is to be executed when conditions are evaluated and actions are performed.

[0029]    In step 72, the compiler 22 then in response to the events symbols header and the state code and using event, time, and memory information from the runtime library generates a runtime code 76.

[0030]    In step 78, controller 11 implements the events in the state models 52, as

instructed in the runtime code 76, in conjunction with the information in the state table 18.

[0031]    Referring now to Figure 3, an example of an implementable state model diagram 100 according to an embodiment of the present invention is shown. The state model frequently contains nested states, which may be generated using a known language in the art, referred to as the Unified Modeling Language.

[0032]    The state model diagram 100 begins at point 102 and entering State 1. State 1 is implemented followed by State 2 and State 3. Notice that each state may have imbedded states such as State 2A and other states as shown. States may refer to other states or consist of commands for performing specific actions, such as powering a component "ON"or "OFF", measuring a voltage level, or measuring a temperature. The specific actions are typically implemented in response to specific conditions. Transitions, represented by arrows 104, exist between states and may have associated events and/or scripted events. This allows state actions to be directly specified on the state model using code fragments. Scripted events are often more complex than other events and have a corresponding script file as to perform multiple actions. Each state model may perform various functions as per application and hardware component 12, as stated above or other desired functions.

[0033]    The present invention provides a general-purpose tool for implementing variously differing state models. The present invention also provides an easy method as for modifying an existing state machine and simplifying storage and implementation of events by recognizing and storing of common events between state models. Furthermore, the present invention provides versatility in the design and implementation of state models by allowing easy and quick modification of event definitions.

[0034]    The above-described apparatus, to one skilled in the art, is capable of being adapted for various purposes and is not limited to the following systems: radiotherapy systems, X-ray imaging systems, ultrasound systems, nuclear imaging systems, computer based systems, and other systems that use state models. The above-described invention may also be varied without deviating from the spirit and scope of the invention as contemplated by the following claims.